(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
12 July 2001 (12.07.2001)

PCT

(10) International Publication Number
WO 01/50335 A1

(51) International Patent Classification[7]: G06F 17/21

(21) International Application Number: PCT/US01/00243

(22) International Filing Date: 4 January 2001 (04.01.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/478,605 6 January 2000 (06.01.2000) US

(71) Applicant *(for all designated States except US)*: **JUST-SYSTEM CORPORATION** [JP/JP]; Brains Park, Tokushima-shi, Tokushima 771-0189 (JP).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **KANTROWITZ, Mark** [US/US]; 1363 Shady Avenue, Pittsburgh, PA 15217 (US). **PELLETIER, Ray** [US/US]; 1617 El Paso Street, Pittsburgh, PA 15206 (US). **SUKTHANKAR, Rahul** [IN/US]; P.O. Box 426131, Cambridge, MA 02142 (US).

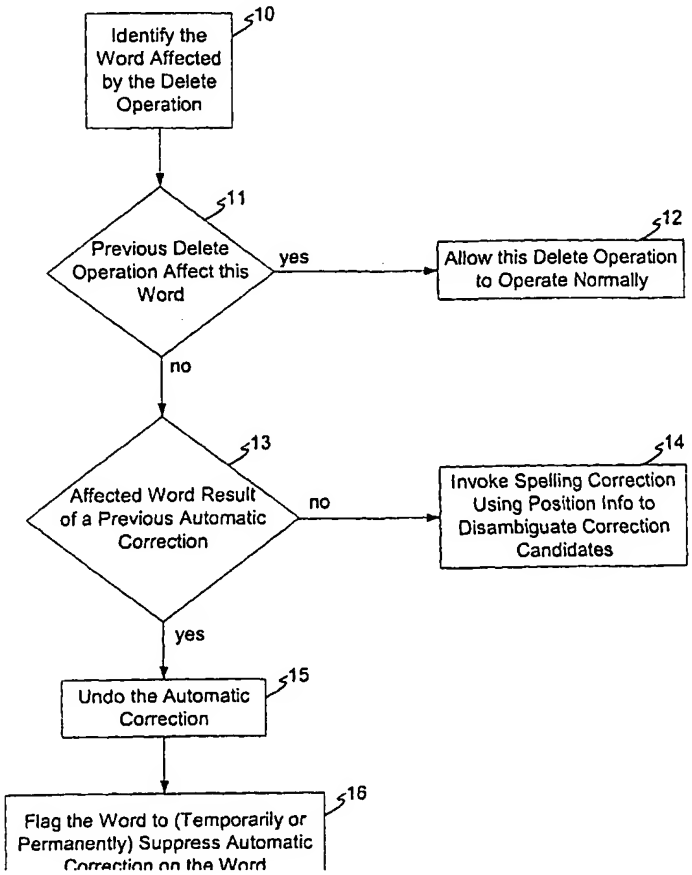
(74) Agents: **BYRNE, Richard, L.** et al.; Webb Ziesenheim Logsdon Orkin & Hanson, P.C., 700 Koppers Building, 436 Seventh Avenue, Pittsburgh, PA 15219-1818 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AT (utility model), AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, CZ (utility model), DE, DE (utility model), DK, DK (utility model), DM, DZ, EE, EE (utility model), ES, FI, FI (utility model), GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (utility model), SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian

[Continued on next page]

(54) Title: UNDOING SPELLING CORRECTION BY OVERRIDING DELETE AND BACKSPACE



(57) Abstract: A computer implemented method for correcting text, including words separated by word break characters, while entering text comprises the steps of upon an event that removes a character in a word (10), looking for an operation for automatically altering the word by adding, deleting or replacing one or more of the remaining characters in the word and if an operation is found, altering the remaining characters in the word according to the operation.

WO 01/50335 A1

patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— *With international search report.*

— *Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

UNDOING SPELLING CORRECTION BY OVERDING DELETE AND BACKSPACE

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates to the entry of text into a computer by any user input system capable of single character operation, for example, a keyboard or a pen stylus on a touch-sensitive writing surface.

2. Description of the Prior Art

When users of a computer notice a spelling or typing error as characters are being entered, they use a delete operation, such as the delete or the backspace function, to manually correct the error, sometimes after repositioning the cursor. In most text editors and word processing programs, the delete key deletes the character following the cursor and the backspace key deletes the character before the cursor. Operators often repeatedly backspace to delete all characters up to and including an error and then resume typing. They sometimes reposition the cursor at or near the error and backspace or delete to correct the error. In the case of pen stroke entry where a character is entered with a prescribed stroke or strokes, as with a hand-held pocket organizer such as the "Palm Pilot", backspacing can be effected by a dash written from right to left.

SUMMARY OF THE INVENTION

The present invention exploits these reflexes to undo automatic spelling correction, automatic word/phrase completion and other automatic substitutions, for example, grammar correction, capitalization correction or automatic formatting, when they modify the word at the current cursor position or to invoke spelling correction, a thesaurus or

other writing tool when the word in question has not been modified.

The methods disclosed herein have application to many text inputting methods for a variety of applications. For example, the methods disclosed herein are useful for word processing software, command line interfaces, text input fields and handwriting recognition systems.

As used herein, "text" comprises words stored as a string of alphanumeric character codes (for example, ASCII codes) separated by codes for word break characters.

As used in the specification and claims, a "word" is a string of characters (usually upper and lower case letters) bound by word break characters, such as a space, a tab, a carriage return or a punctuation mark.

Unless otherwise specifically set forth, the term "delete operation" as used in the specification and claims refers to striking a key that either removes a character following the cursor (often marked "Del" on the keyboard) and/or removes the character preceding the cursor (often marked "Backspace" on the keyboard). It also refers to a pen stroke that is interpreted as a backspace/delete with a handwriting text entry system.

As used in the specification and claims, an "operation" refers to a rule-based procedure, an algorithm or a submission to a neural network. Therefore, for the purposes of this application, an "operation" is any procedure that accepts a word or phrase as input and produces a word or phrase as output. The output is typically substituted for the input text and can be referred to as a "substitution" or "correction".

Briefly, according to this invention, there is provided a computer implemented method for correcting text while it is being entered. According to this method, upon a delete operation, a determination is made whether: a) the word affected has not been previously corrected or completed by an operation for automatic alteration and is currently misspelled, b) the word affected has been previously corrected or completed by an operation for automatic alteration or c) neither a) nor b).

In step a), if the word has not been previously corrected/completed and the word was misspelled before the removal of a character by the delete operation, spelling correction will be invoked. The spelling correction algorithm will initiate a search to determine whether an operation to invoke a unique spelling correction can be applied. If so, that operation will be applied. Otherwise, alternative substitutions (such as other spellings, synonyms, antonyms, hypernyms and hyponyms) may be selected.

The spelling correction algorithm can be a rule-based spelling correction algorithm or any other spelling correction procedure so long as the procedure transforms an input word into an output word. The search to identify an applicable operation involves testing the different operations to see which, if any, applies to the input word. In a rule-based spelling correction algorithm, testing a rule involves checking whether the rule's conditions match the input word (i.e., whether the pattern of the rule matches the input word). In spelling correction algorithms that are not rule-based, testing an operation involves applying the operation to a copy of the input word and

comparing the resulting word to the input word. If the two are different, then the operation is applicable.

In step b), if the delete operation affects a word that has already been automatically corrected or completed, the previously implemented alteration will be undone. Preferably, the word for which an automatic alteration of the word has been overridden (undone) is marked so that the word can be treated differently upon the invocation of other word processing procedures. For example, a batch processing spell correction procedure will not attempt to correct a word for which automatic correction was undone.

In step c), if the delete operation invoked extra functionality (override of automatic alteration or invocation of spelling correction), subsequent keystrokes will operate normally for the affected word. In situations where the delete operation does not invoke extra functionality, the delete operation will function as a normal deletion (i.e., remove a character).

A delete operation may also be applied to selected (or blocked) text that may be comprised of more than one word. Selected text is text that is blocked and highlighted using some combination of keystrokes and/or the mouse. If the delete operation is performed on a selected word, spelling correction will be invoked. If the delete operation is performed on a partially selected word, spelling correction will be invoked using the cursor location as a hint to the problem with the word. Lastly, if the delete operation is performed on a selected phrase, grammar correction will be invoked.

BRIEF DESCRIPTION OF THE DRAWINGS

Further features and other objects and advantages will become apparent from the following detailed description made with reference to the drawings in which:

Fig. 1 is a flow diagram illustrating the general process according to this invention; and

Figs. 2(a) and 2(b) are flow diagrams for illustrating a computer program object for implementing one embodiment of this invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

With a computer word processing program, the words are stored in memory as codes and are also displayed as text on a monitor or other display. The word processing program simultaneously displays a cursor and points to the location in memory corresponding to the location of the cursor on the display. This is typical of word processing programs.

This invention has application to many text inputting methods. Not only will the invention apply to word processing or text editing programs wherein characters are entered from the keyboard, but also to command line interfaces, text input fields and typewriters, among other applications. Additionally, this invention has application to small computers where characters are entered by pen strokes with a stylus on a touch sensitive pad such as the "Palm Pilot". The "Palm Pilot" is a pocket organizer manufactured by U.S. Robotics Corporation. It has a touch sensitive display screen which responds to the touch of a stylus. It also has a writing area where pen strokes are made which can be interpreted as alphanumeric characters, punctuation marks and spaces. The pen strokes are designed

to closely resemble those of the regular alphabet. A character entered in the writing space is entered in memory and displayed as text on the display screen.

Referring to Fig. 1, an initial step 10, in the method according to this invention, is the identification of the word affected by the deletion. In the case of keyboard text entry, the backspace or delete keys can be applied to the letters within the word or to word break delimiters. If the delete operation removes a letter within a word, that word is deemed the affected word. If the delete operation removes a word break delimiter, the direction of the deletion determines the affected word. A backward deletion caused by a backspace at the end of the word affects the word to the left of the cursor. A forward delete affects the word to the right of the cursor. Preferably, the deleted character is saved for use in a later step.

A next step 11, in the method according to this invention, is to determine if the previous delete operation affected the word. If so, at step 12, the delete operation is allowed to operate normally.

A next step 13 in the method is to determine if the affected word is the result of a previous automatic correction. If the word has not been modified by automatic spelling correction or other automatic substitutions, then at step 14, spelling correction is invoked and provided with information about the letters affected by the delete operation to be used as a hint to the position of the possible error. The remaining letters in the word are also used to identify a correction. If the spelling correction algorithm provides a unique correction, that correction is

executed. Otherwise, correction may be deferred until additional letters are keyed that may identify a unique correction which is then executed. In one embodiment, a pop-up menu of alternate choices may be displayed allowing the user to select the correction to be executed. In addition to possible correction candidates, the choices may include synonyms, antonyms, hypernyms, hyponyms and other words semantically related to the word affected. Preferably, user-configurable parameters control whether the choices include synonyms, antonyms and the like, whether subsequent deletions cycle through the choices or a menu of the choices pops up, whether only unique corrections occur (with deletions operating normally in cases of ambiguous correction) and so forth.

At step 15, if the word has been modified by a previous automatic correction, the correction is undone and subsequent keystrokes are allowed to operate normally upon the word. After the operator enters a word delimiter or uses cursor movement to end modification of the word, the word is flagged at step 16 to suppress any further automatic correction of the word, such as by a batch spelling correction procedure.

According to a preferred embodiment, external deletion, that is, deletion of word break characters adjacent a word, may be handled according to various setable parameters. For example, external delete may undo automatic correction. A following spacebar stroke may redo the correction. Control-space may enter a word without redoing the automatic correction. This permits the user to switch back and forth between two views of the word using the spacebar and a delete key. In an alternate preferred

embodiment, external delete undoes automatic correction and temporarily suppresses the automatic correction so that the spacebar does not invoke automatic correction on the word. This prevents the automatic correction process from affecting the word when the user resumes typing.

A special case may be required for embodiments that permit the method to be applied to selected text. Selected text is text that is blocked and highlighted using some combination of keystrokes and/or the mouse. If the selected text consists of one word, spelling correction is invoked on the word using the most specific spelling substitution available. If the selected text comprises multiple words, grammar correction is invoked. If the selected text consists of a part of a word, the spelling correction algorithm is invoked on the word using the selected text as a hint to the problem with the word.

According to the preferred embodiments of this invention, the user can set a parameter to control which behaviors are invoked in each situation. For example, instead of undoing the previous correction, the user could set a parameter to have the delete operation pop up a menu of high-probability alternate corrections. It may be desirable to use less than all of the functionality described above for some tasks. Thus, for example, the automatic undoing of corrections may be turned off or spelling correction may be turned off.

Various spelling correction algorithms may be used with this invention. However, the preferred spelling correction algorithm is based upon rules for matching unlikely n-grams with likely replacement n-grams. (An n-gram is a string of characters that may comprise all or

part of a word.) The rules within the spelling correction algorithm are expressions that match pieces of correctly spelled words and replace incorrect pieces in the text. Examples of possible rules are as follows:

oualy → ously
mnet → ment
fuly → fully
ierd → eird
eif → ief

The preferred spelling correction algorithm is described in considerable detail in a co-pending U.S. Patent Application Serial No. 09/312,229, filed May 14, 1999 entitled "Method for Rule-Based Correction of Spelling and Grammar Errors", and assigned to the same assignee as this patent application. However, the spelling correction algorithm can be any spelling correction procedure, so long as the procedure transforms an input word into an output word. See Kukich, Karen, "Techniques for Automatically Correcting Words in Text", ACM Computing Surveys, Vol. 24, No. 4, December 1992.

In a preferred embodiment, the spelling correction algorithm uses information about the position of the character deleted to disambiguate between several possible corrections. This only works when the affected letters are internal to the word and so represent a clear signal that the user's intent is to correct those letters. For example, if the user entered "amoung", the most likely correction is "amount". However, if the user deletes the "u", then "among" is the most likely correction. This is

an example where position information indicates that deletion should be treated as regular deletion and not as a signal to change the spelling of the word.

Corrections should only be made if there exists a unique correction. The present invention finds more unique corrections as it obtains more precise information about the nature of the problem, such as the position of the error. This extra information allows the spelling correction algorithm to restrict the set of candidate corrections allowing more errors to be corrected automatically. If the user moves the cursor into the middle of a word and begins deleting characters, this indicates the position of the problem with the word. For example, if the user moves the cursor just after "ie" in "receive", it is likely that the user will replace those characters. Likewise, if the operator deletes all characters up to a point and resumes typing, this implies that the prefix of the word is complete and again indicates the position of the problem. For example, if the user types "quiet", deletes the "et" and types a "t", one can assume that the next character will be an "e", not a space. The fact that the user originally typed "quiet" with an "e" increases the probability that this was a transposition error as opposed to an insertion error. Thus, the most likely correction is "quite", not "quit".

Figs. 2(a) and 2(b) together are a programming flow diagram describing a program to implement the process using a rule-based spelling correction algorithm according to this invention. Starting at Fig. 2(a), the State is initialized at step 20. At steps 21 and 23, events are processed and if a Key event is detected at step 23,

processing begins. If a rule has not been applied (in other words, the State is Looking for Rule) and the Key is Delete as determined at step 25, then at step 26 a rule is sought by using the characters remaining in the word. If a rule is found for correcting the word, it is saved as a Deferred Rule at step 27. The rule is not applied because subsequent deletions and/or character additions may provide a better rule. Program flow from here awaits the next event. If the State is Looking for Rule and the Key is not Delete at step 25, the flow moves to B in Fig. 2(b), which is the automatic spell correction routine.

With continued reference to Fig. 2(a), if a Key event takes place, the State is Applied Rule (a rule has been applied to the word) as determined at steps 27 and 28 and the Key is Delete as determined at step 29, then at step 30 the rule is undone and the State is reset at step 31. This enables an automatic correction to be overridden. If at step 29 the Key is not Delete, the State is reset at step 32.

Referring to Fig 2(b), starting at B, the State is Looking for Rule and the Key is not Delete as determined by steps 24 and 25. If the Key is found to be Word Break at step 33, a search is made for a rule at step 34. If a rule is found at step 35, it is applied at step 36, the State is set to Rule Applied (step 46) and the rule number is saved (step 47) so that the rule can be reversed. Then the next event is awaited.

If a rule is not found at step 35, but a rule had been deferred as determined at step 37, the Deferred Rule is applied at steps 36, 46 and 47. If no rule is found and

no rule has been deferred, the State is reset at step 38 and the next event is awaited.

If the State is Looking for Rule and the Key is not Delete or Word Break (i.e., the Key is Character, the word being entered) as determined at steps 24, 25 and 33, at step 40 a search is made for a rule based upon the characters already entered in the word. If a rule is found as determined at step 41, but a more specific rule is possible based on the addition of more characters to the word as determined at step 42, then the rule is saved at step 43 as a Deferred Rule and the next event is awaited. If no rule is found at step 41 but a Deferred Rule is being held as determined at step 44, the Deferred Rule is applied at step 36, the State is set to Rule Applied at step 46 and the rule number is saved at step 47; then the program awaits the next event. If no rule is found at step 41 and no Deferred Rule is being held, the State is reset at step 45 and the next event is awaited.

The present invention has been implemented with the "Palm Pilot" for spelling correction assisted handwriting recognition. The "Palm Pilot" is a hand-held computer with a handwriting interface. On the face of the display of the "Palm Pilot", there is provided a space for hand forming characters that are then translated and displayed. The spelling correction system intercepts the characters entered by the user and uses a variety of rules and heuristics to correct errors in the input. For example, if the handwriting recognition software that comes with the "Palm Pilot" misreads a letter, an automatic spelling correction substitutes the correct letter. Sometimes the automatic substitution, according to this

invention, produces an erroneous change. For example, the user might insert an acronym that appears to the "Palm Pilot" to be an error. The "Palm Pilot" will automatically correct the presumed error. The user may then use a delete stroke to remove the automatic correction. The delete stroke is interpreted as a command to undo the automatic change. Subsequent delete strokes operate normally. If the user enters the same sequence of letters again that led to the automatic substitution, the automatic substitution is suppressed.

The present invention has also been implemented in GNU-Emacs which is a word processing program with keyboard input.

The following computer code in the C language sets forth portions of a function for implementing the correction of handwriting and spelling errors in the embodiment of this invention implemented with the "Palm Pilot".

```
Boolean HackFldHandleEvent (FieldPtr fld, EventPtr eventP)
{
     enum events eType = eventP->eType;

     if (eType == keyDownEvent) {
          return StepChar (fld, eventP);
     }

     if (eType == penDownEvent) {
          return StepPen (fld, eventP);
     }
     else
     {
          return NextFldHandleEvent(fld, eventP);
     }
}

Boolean StepChar (FieldPtr fld, EventPtr eventP)
{         result = StepCharAux (fld, eventP, globalsP, trieP);
          return result;
}

Boolean StepPen (FieldPtr fld, EventPtr eventP)
{
           StepPenAux (fld, eventP, globalsP);
}

Boolean StepCharAux (FieldPtr fld, EventPtr eventP,
                     KSGlobals *globalsP, Word *trieP)
{
     RuleDesc ra;
```

```
Word specificity = 2;
Boolean result = true;
char c = eventP->data.keyDown.chr;

switch (globalsP->state)
{
case KSS_LOOKING_FOR_RULE:
     result = NextFldHandleEvent(fld, eventP);
     if (result)
     {
          if (globalsP->rule.index != 0)
          {
               specificity = toSpecificity(toRule(&globalsP->rule));
          }

          switch (charType(c))
          {
          case C_DELETE:
               DecCC(globalsP);
               SetRule(globalsP, 0);
               if (LookupRule (globalsP, trieP, &ra, fld, specificity))
                    SetRule(globalsP, &ra);
               break;

          case C_WORDBREAK:
               IncCC(globalsP);

               if (LookupRule (globalsP, trieP, &ra, fld, specificity))
               {
                  . ApplyRule (globalsP, fld, toRule(&ra), -1);
                   SetState(globalsP, KSS_RULE_APPLIED);
```

```
        }
        else if (globalsP->rule.index != 0)
        {
             ApplyRule (globalsP, fld, toRule(&globalsP->rule), -1);
            SetState(globalsP, KSS_RULE_APPLIED);
        }
        else
        {
             ClrCC(globalsP);
            SetRule(globalsP, 0);
        }
        break;

    case C_TARGET:
        IncCC(globalsP);

        if (!LookupRule (globalsP, trieP, &ra, fld, specificity))
        {
             if (globalsP->rule.index != 0)
             {
             ApplyRule(globalsP, fld, toRule(&globalsP->rule), -1);
             SetState(globalsP, KSS_RULE_APPLIED);
             }
        }
        else
        {
             SetRule(globalsP, &ra);
             if (!LookupPartialRule (globalsP, trieP))
             {
                  ApplyRule(globalsP, fld, toRule(&ra), 0);
             SetState(globalsP, KSS_RULE_APPLIED);
```

```
            }
        }
        break;

    case C_OTHER:
        SetRule(globalsP, 0);
        ClrCC(globalsP);
        break;
    }
}
break;

case KSS_RULE_APPLIED:
    switch (charType(c))
    {
    case C_DELETE:
        UndoRule(globalsP, fld, toRule(&globalsP->rule));
        ClrCC(globalsP);
        break;

    case C_TARGET:
        result = NextFldHandleEvent(fld, eventP);
        ClrCC(globalsP);
        if (result) IncCC(globalsP);
        break;

    default:
        result = NextFldHandleEvent(fld, eventP);
        ClrCC(globalsP);
        break;
    }
```

```
        SetRule(globalsP, 0);
        SetState(globalsP, KSS_LOOKING_FOR_RULE);
        break;
    }
    return result;
}
```

The program from which the above programming functions are taken is event driven. Thus, the object HackFldHandleEvent has for its purpose to handle events, such as the keyDownEvent, which is entry of characters to the field. The penDownEvent occurs when the pen is used to change the cursor position for example.

A keyDownEvent passes control to the StepChar function which, among many other tasks, passes control to the StepCharAux function where the processing of characters and codes to effect the purposes of the invention takes place.

It should be understood that certain global constants represent values that define several states that determine how the StepCharAux object reacts to the entry of a particular character or code. The program is initially in the State named by the constant KSS_LOOKING_FOR_RULE. The State is also set to KSS_LOOKING_FOR_RULE if a misspelling has been identified in a word being entered and a potential rule has been found, but the application of the rule has been deferred while looking for a more specific rule. The State is set to KSS_RULE_APPLIED if a rule has been applied to a word.

Three other global constants characterize the type of character just entered: C_WORDBREAK identifies a word break character, such as a space; C_TARGET identifies a character within a word; and C_DELETE identifies a delete code.

The way in which the function StepCharAux responds to a new character or code depends upon the States defined by the five constants explained above. The function StepCharAux is structured as nested switch

statements. Of course, the method, according to this invention, can be arbitrarily implemented in hardware or software.

Having thus described our invention with the detail and particularity required by the Patent Laws, what is desired to be protected by Letters Patent is set forth in the following claims.

WE CLAIM:

1. A computer implemented method for correcting text, including words separated by word break characters, while entering text, said method comprising the steps of:

a) upon an event that removes a character in a word, looking for an operation for automatically altering the word by adding, deleting or replacing one or more of the remaining characters in the word; and

b) if an operation is found, altering the remaining characters in the word according to the operation.

2. The method according to claim 1, wherein the event that removes a character in the word will undo a previously implemented automatic alteration of the word.

3. The method according to claim 2, further comprising the step of marking the word for which an automatic alteration of the word has been overridden so that the word can be treated differently upon the invocation of other word processing procedures.

4. The method according to claim 2, wherein the event that removes a character in the word will undo a previously implemented word completion.

5. The method according to claim 3, wherein the event that removes a character in the word will undo a previously implemented user performed alteration of the word.

6. The method according to claim 1, further comprising the step of using the position of the character removed to choose an operation for altering the word.

7. A computer implemented method for entering text including words separated by word break characters, said method comprising the steps of:

a) upon an event that removes a character in a word, determining whether an operation for automatically invoking a writing tool can be chosen; and

b) if an operation is chosen, invoking the tool.

8. The method according to claim 7, wherein the writing tool is a thesaurus.

9. The method according to claim 8, wherein the thesaurus allows for the selection of one of alternative choices for substitution of the word.

10. A computer implemented method of correcting text, including words stored as a string of alphanumeric character codes separated by word break character codes, while entering text, said method comprising the steps of:

a) on a deletion event, determining whether a word character or a word break character was deleted;

b) if a word break character was deleted, determining which word is affected according to the direction of the deletion;

c) based upon the determination in steps a) and b), seeking an operation for automatically altering the affected word; and

d) if an operation is found, altering the affected word according to the operation.

11. The method according to claim 10, wherein an operation is sought based upon the remaining characters in the word.

12. The method according to claim 10, wherein no operation is sought if an operation has already been applied to alter the word.

13. The method according to claim 12, wherein a previous automatic alteration of the word is undone and further additions or deletions to the word are normal.

14. The method according to claim 13, wherein a previous user performed alteration of the word is undone and further additions or deletions to the word are normal.

15. The method according to claim 10, wherein if an operation has not already been applied to alter the word, upon entry of a delete code, an operation is sought based upon the remaining characters in the word and applying the operation if the possibility does not exist that a more specific operation may apply following entry of the next character in the word, but if the possibility of a more specific operation exists, holding the operation as a deferred operation.

16. The method according to claim 15, wherein on entry of a word break character, an operation is sought based upon the characters in the word and if found applied, but if not found, then applying an operation that has been deferred.

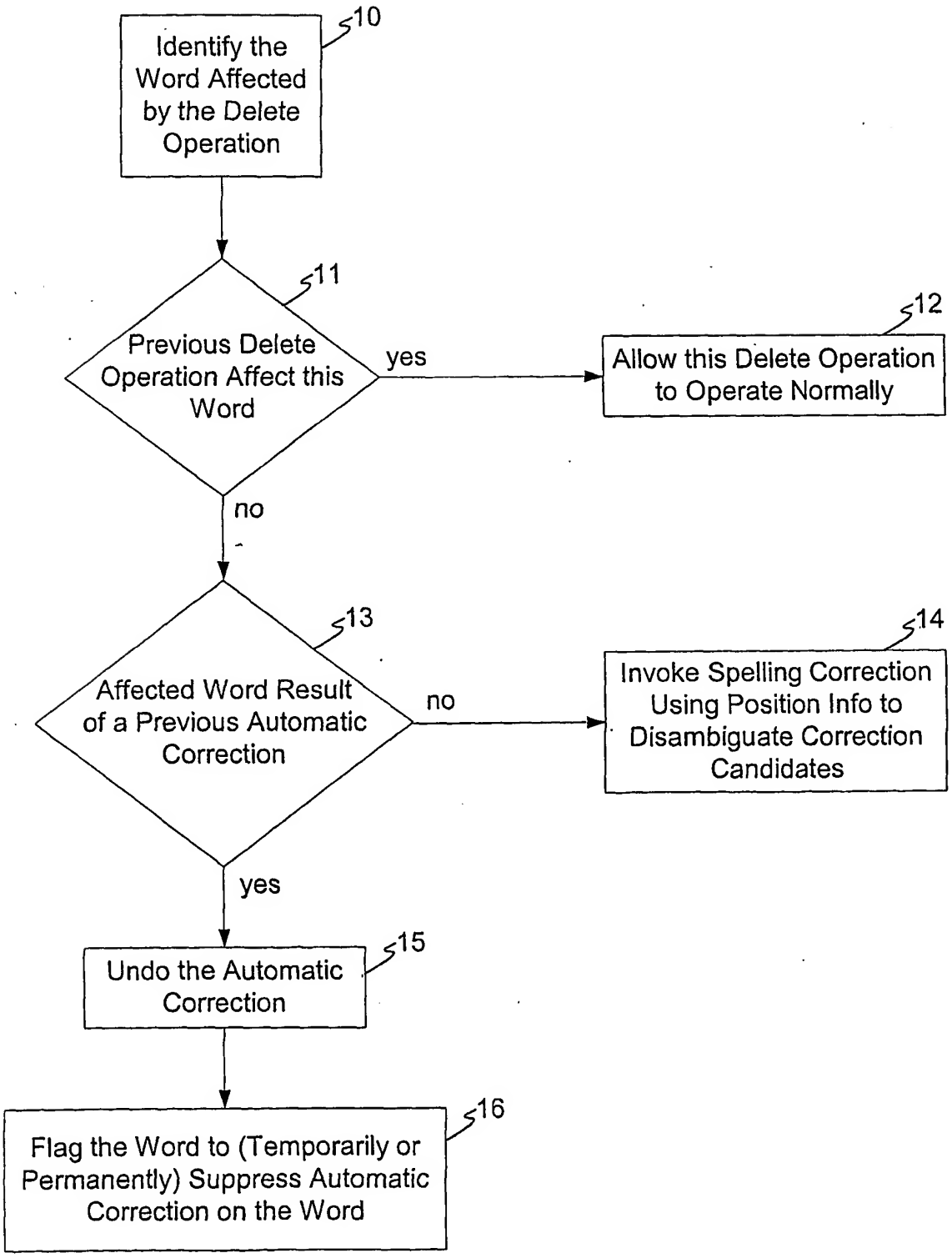

Identify the Word Affected by the Delete Operation
10
Previous Delete Operation Affect this Word
11
yes
Allow this Delete Operation to Operate Normally
12
no
Affected Word Result of a Previous Automatic Correction
13
no
Invoke Spelling Correction Using Position Info to Disambiguate Correction Candidates
14
yes
Undo the Automatic Correction
15
Flag the Word to (Temporarily or Permanently) Suppress Automatic Correction on the Word
16

FIG. 1

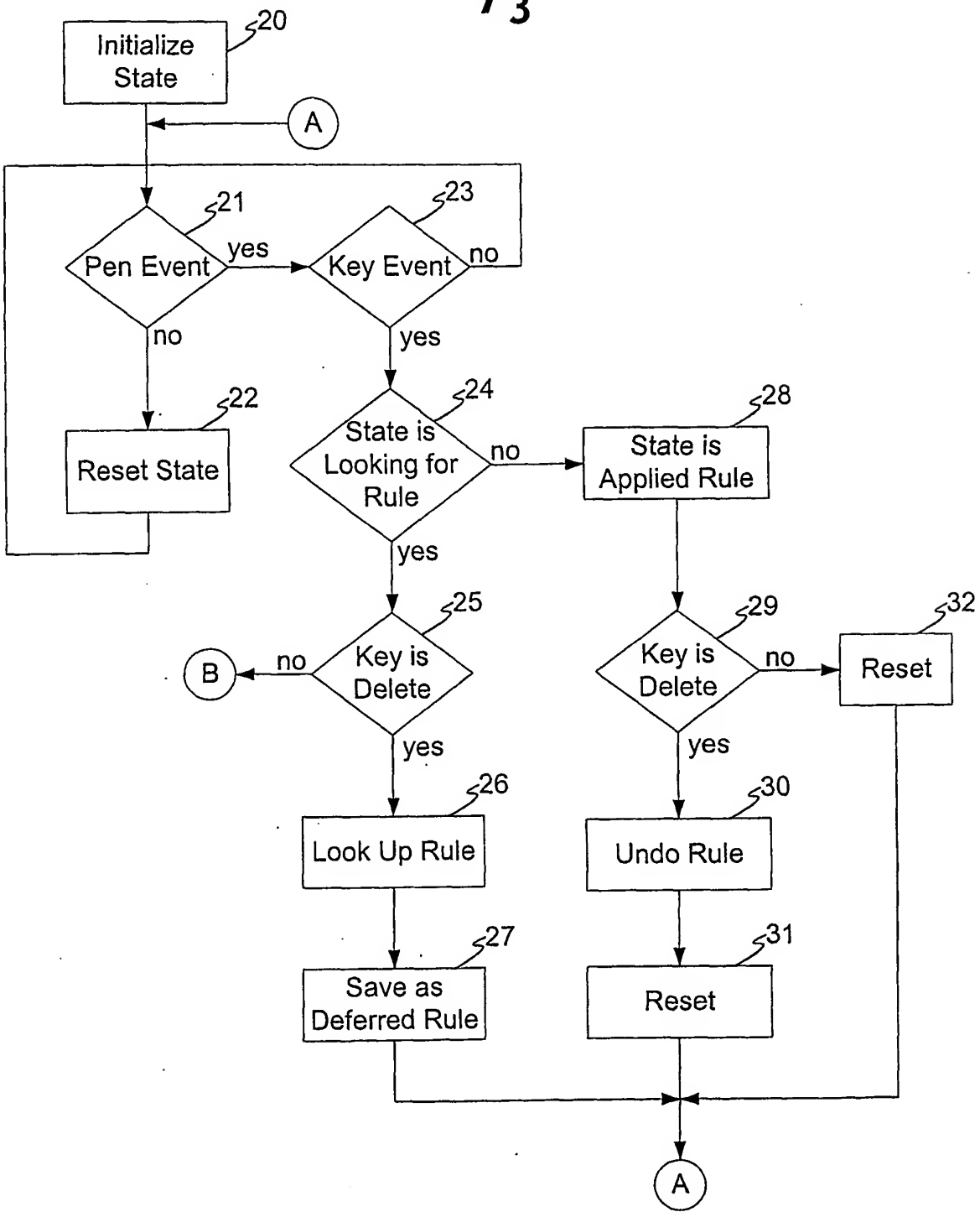

Initialize State
20
A
21
Pen Event
yes
23
Key Event
no
no
yes
22
Reset State
24
State is Looking for Rule
no
28
State is Applied Rule
yes
25
B
no
Key is Delete
29
Key is Delete
no
32
Reset
yes
yes
26
Look Up Rule
30
Undo Rule
27
Save as Deferred Rule
31
Reset
A

FIG. 2(a)

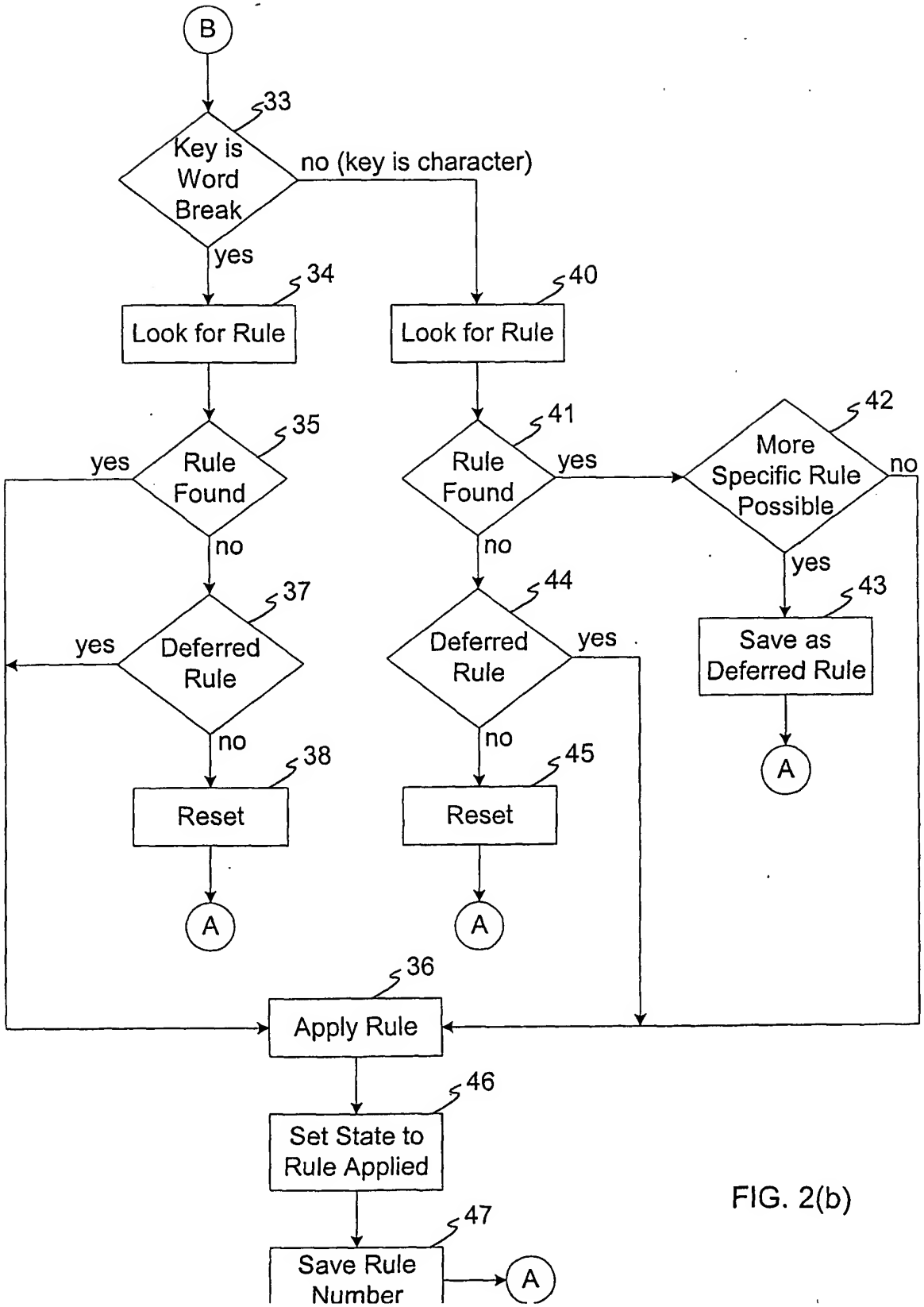

B
33
Key is Word Break
no (key is character)
yes
34
Look for Rule
40
Look for Rule
35
Rule Found
yes
no
41
Rule Found
yes
no
42
More Specific Rule Possible
no
yes
37
Deferred Rule
yes
no
44
Deferred Rule
yes
no
43
Save as Deferred Rule
38
Reset
45
Reset
A
A
A
36
Apply Rule
46
Set State to Rule Applied
47
Save Rule Number
A

FIG. 2(b)

INTERNATIONAL SEARCH REPORT

Interr ation No.
PCT/US01/00243

A. CLASSIFICATION OF SUBJECT MATTER
IPC(7) : G06F 17/21
US CL : 707/531, 532, 533
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 707/531, 532, 533

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
USPAT, US-PGPUB, IBM TBD, EPO, JPO, DERWENT

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,907,839 A (ROTH) 25 May 1999, column 3, lines 66-67, column 4, lines 1-67, column 5, lines 1-67, column 6, lines 1-22.	1-16
Y	US 5,940,847 A (FEIN et al) 17 August 1999, column 1, lines 20-67, column 2, lines 1-67, column 3, lines 1-67, column 4, lines 1-53.	1-16
Y	US 5,812,863 A (ISHIKAWA) 22 September 1998, column 1, lines 7-67, column 2, lines 1-67, column 3, lines 1-67, column 4, lines 1-67, column 5, lines 1-67, column 6, lines 1-67, column 7, lines 1-10.	1-16
Y	US 5,765,180 A (TRAVIS) 09 June 1998, column 1, lines 17-67, column 2, lines 1-26.	1-16
Y	US 5,678,053 A (ANDERSON) 14 October 1997, column 3, lines 10-67, column 2, lines 1-62.	1-16
Y	US 5,649,222 A (MOGILEVSKY) 15 July 1997, column 1, lines 9-67, column 2, lines 1-20.	1-16
Y	US 5,576,955 A (NEWBOLD et al) 19 November 1996, column 1, lines 10-67, column 2, lines 1-54.	1-16
Y	US 5,437,036 A (STAMPS et al) 25 July 1995, column 1, lines 14-67, column 2, lines 1-67, column 3, lines 1-67, column 4, lines 1-15.	1-16

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
24 April 2001 (24.04.2001)

Date of mailing of the international search report
16 MAY 2001

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. (703)305-3230

Authorized officer
HEATHER HERNDON R. Mattingly
Telephone No. (703) 305-3900

Form PCT/ISA/210 (second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

Inter nal application No.
PCT/US01/00243

C (Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,367,453 A (CAPPS et al) 22 November 1994, column 1, lines 7-67, column 2, lines 1-62.	1-16
Y	US 5,974,194 A (BARKER et al) 27 November 1990, column 1, lines 18-67, column 2, lines 1-67, column 3, lines 1-31 .	1-16